

CLAIMS

1. A method of creating a core file derived from an application crash, said application belonging to a source platform and being run on a target platform comprising the steps of:
 - emulating the application on the target platform;
 - responding to the crash occurring by detecting whether the crash corresponds to the failure of a process corresponding to the emulated application;
 - mediating the crash process by intercepting information relating to the failure of the emulated application and writing a core file corresponding to the failure of the emulated application running on the source platform.
2. A method of creating a core file as claimed in claim 1 wherein the mediation process includes the steps of:
 - suspending all processes except for the process handling the creation of the core file; and
 - writing the contents of the emulated applications core file to non-volatile memory.
3. A method of creating a core file as claimed in claim 1 wherein the mediation process includes the step of writing out data relating to the emulation state of the application to the core file.
4. A method of creating a core file as claimed in claim 1 wherein the mediation step includes the step of writing out data relating to the process, data segments, stack segments, shared memory segments, and MMAP memory segments, to the core file.

5. A method of creating a core file as claimed in claim 4 wherein the mediation step further includes one or more of the steps of writing out information identifying the core version, the process which caused the crash, the host platform kernel version to which the core file belongs, and the name of the application which produced the core file.
6. A method of creating a core file as claimed in claim 2 including the steps of setting up the core file book-keeping parameters including one or more of setting permissions for the core file, specifying the core file name, determining the maximum allowable size for the core file, creating an empty core file, and determining the number of writable segments to be written to the core file.
7. A method of creating a core file as claimed in claim 6 further including the step of writing out a specified file header to the core file.
8. A method of creating a core file as claimed in claim 1 wherein the mediation step is invoked in response to one or more specified signals being received from the operating system of the target platform.
9. A method of creating a core file as claimed in claim 1 wherein the emulation step includes monitoring book-keeping information required to mediate a crash and creating of a core dump file relating to the emulated application.
10. A method of creating a core file as claimed in claim 9 wherein the book-keeping step includes monitoring one or more data relating to the state of the emulated application.
11. A method of creating a core file as claimed in claim 10 wherein the data relating to the state of the emulated application includes one or more of the start address of the data segment of the application, the size of the data segment of the application, and system calls made by the application which specify the state of the emulated application.
12. A method of creating a core file as claimed in claim 1 including the step of implementing a signal handler adapted to intercept the signals which, on receipt of said signals, would cause a core dump in the target platform.
13. A binary translator adapted to operate in accordance with the method of claim 1.

14. A computer adapted to operate in accordance with the method of claim 1.
15. A memory storing instructions specifying a method of operating a computer as claimed in claim 1.